CLAIMS

1. A method for generating timing constraints for use by a digital logic optimization synthesis tool to optimize a digital circuit design by applying said timing constraints thereto, comprising the steps of:

temporarily replacing flip-flops of said digital circuit with buffers having a negative delay whose magnitude is approximately a desired clock cycle time of said digital circuit; and

temporarily breaking cycles in said digital circuit using flip-flops having an infinite or quasi-infinite clock frequency.

2. The method of Claim 1, wherein said timing constraints are used for a digital circuit that is to be retimed after optimization.

3. The method of Claim 1, wherein following optimization by said synthesis tool, the temporary changes of Claim 1are reverted, and retiming is performed on said digital circuit.

4. A method for generating timing constraints, comprising the steps of:

describing a digital circuit using a standard HDL;

constructing said digital circuit from said HDL description; and

replacing flip-flops in said digital circuit with negative delay elements.

5. The method of Claim 4, wherein said negative-time elements are implemented by buffers having a delay -T, where T is a delay equal to a flip-flop's clock period less a typical flip-flop delay.

6. The method of Claim 4, wherein if said digital circuit contains cycles, then performing the step of breaking said cycles by inserting flip-flops clocked by clocks all having a period of substantially zero.

7. The method of Claim 6, wherein cycles are only broken on backward paths.

8. The method of Claim 4, wherein clocks and registers constructed have the property of slack equivalence, wherein optimization goals at each gate are substantially the same as they would be if registers were already optimally distributed.

9. The method of Claim 5, wherein the actual value of T is set to a clock period of a flip-flop being replaced.

10. The method of Claim 4, further comprising the step of:

using a buffer to replace a flip-flop, said buffer having a typical load capacitance, representing an average or weighted-average load capacitance taken over inputs of all gates and flip-flop D pins in a target technology library.

11. The method of Claim 5, further comprising the step of:

describing a value of T using a capacitance/delay curve representing a composite of gates in a target technology library, Q pins of flip-flops in said target technology library, and a series of increasingly powerful buffer trees;

wherein said curve is first computed, then it is offset by setting a delay corresponding to typical load capacitance to –T;

whereby a larger capacitive load results in a longer delay; and

whereby if a near-zero load is imposed a delay is –(T + t), where t is a (positive) difference in delay between a typical load and a lesser load.

12. The method of Claim 4, further comprising the steps of:

after logic optimization, reinstalling registers in place of negative-delay elements;

removing all zero-clocked cycle-breaking flip-flops;

applying a retiming algorithm; and

after retiming, performing a second logic optimization pass to fine-tune said retimed design.

13. A method for generating timing constraints, comprising the steps of:

describing a digital circuit using a standard HDL;

constructing said digital circuit from said HDL description; and

replacing flip-flops in said digital circuit with point-to-point timing constraints.

14. A method for generating timing constraints, comprising the steps of:

describing a digital circuit using a standard HDL;

constructing said digital circuit from said HDL description; and

constructing a system of clocks and artificial flip-flops that creates slack equivalence by the following steps:

assigning inputs to a stage 0;

computing a length of a longest path from any input to each output;

assigning said outputs to stages as denoted by the longest path length;

adding skips (1T delays) where alternative paths are shorter than the longest path;

establishing a series of clocks { C_0, C_1, C_2, ...) with a long clock period and a phase delay of T between successive members of said series;

for each output and input, constructing a dummy flip-flop; and

clocking said dummy flip-flop with a clock whose subscript is a stage to which the input or output has been assigned.

15. A method for generating timing constraints, comprising the steps of:

describing a digital circuit using a standard HDL;

constructing said digital circuit from said HDL description; and

breaking cycles by inserting flip-flops clocked by clocks all having a period of substantially zero.

16. The method of Claim 15, wherein said step of breaking cycles is performed by using the following algorithm:

```
mark all forward paths, e.g. by using depth-first search
while there are unbroken cycles
    choose a cycle C
    traverse C looking for an edge E such that E is not marked as forward
    if such an E is found, break E
    else choose an E such that it does the least damage when broken
        and break E
end.
```

17. The method of Claim 15, wherein said step of breaking cycles is performed by breaking only redundant edges if they are present in a cycle; wherein edges are broken that can be removed from a graph without disconnecting any previously connected input/output pairs.

18. The method of Claim 17, wherein if multiple redundant edges exist, then performing the steps of:

examining logic along said redundant edges; and

choosing a redundant edge having the least logic;

wherein if no redundant edge exists, then it is assumed that there is a backward edge that can be broken but, such assumption is false, an edge is chosen at random to break said cycle.